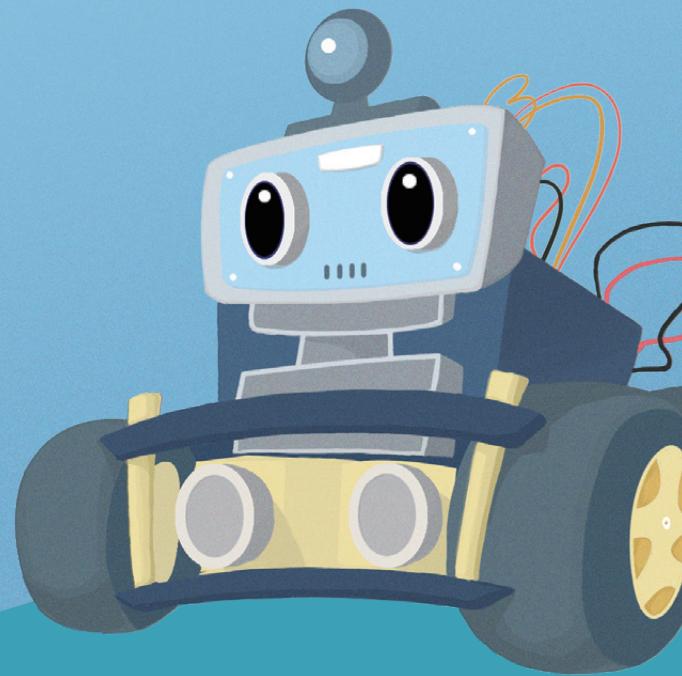


8

SMART ROBOT CAR V4.0 WITH CAMERA



DIY and Graphical
Programming
on APP





Introduction:

- + If you find it difficult and boring to write programs on the Arduino software and are tired of the boring theoretical study in the previous part, then in this lesson, we will teach you how to use the vivid, fun and easy graphical programming and DIY mode on the supporting APP to control the car to achieve the effect and function you want.



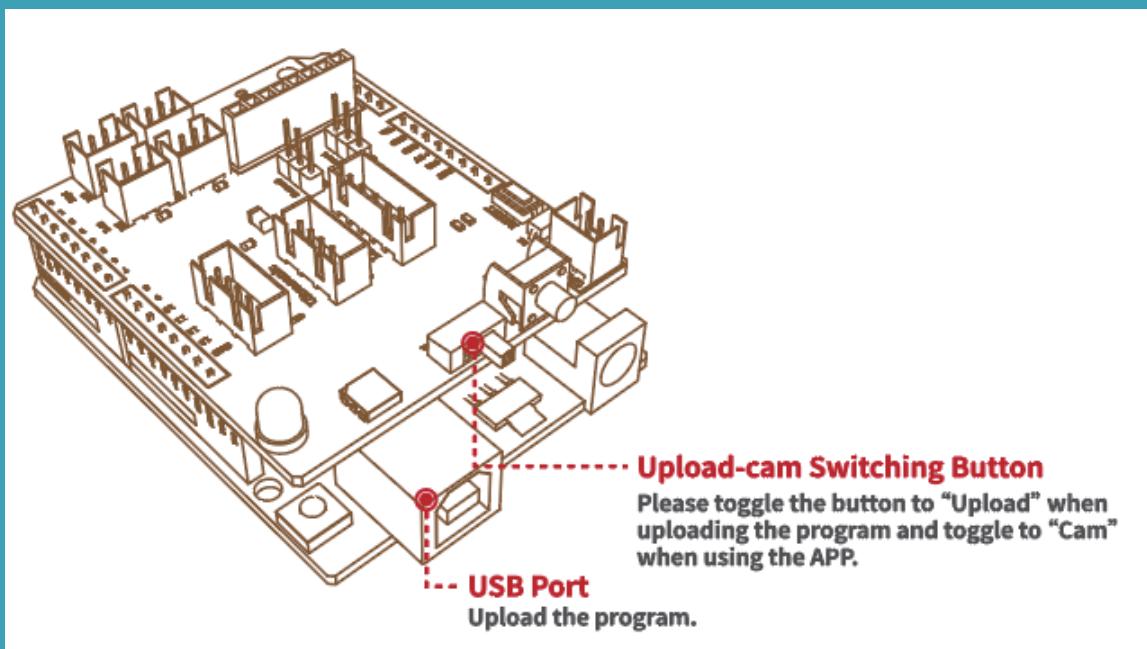
Preparation:

- + A Smart Robot Car V 4.0 (with battery)

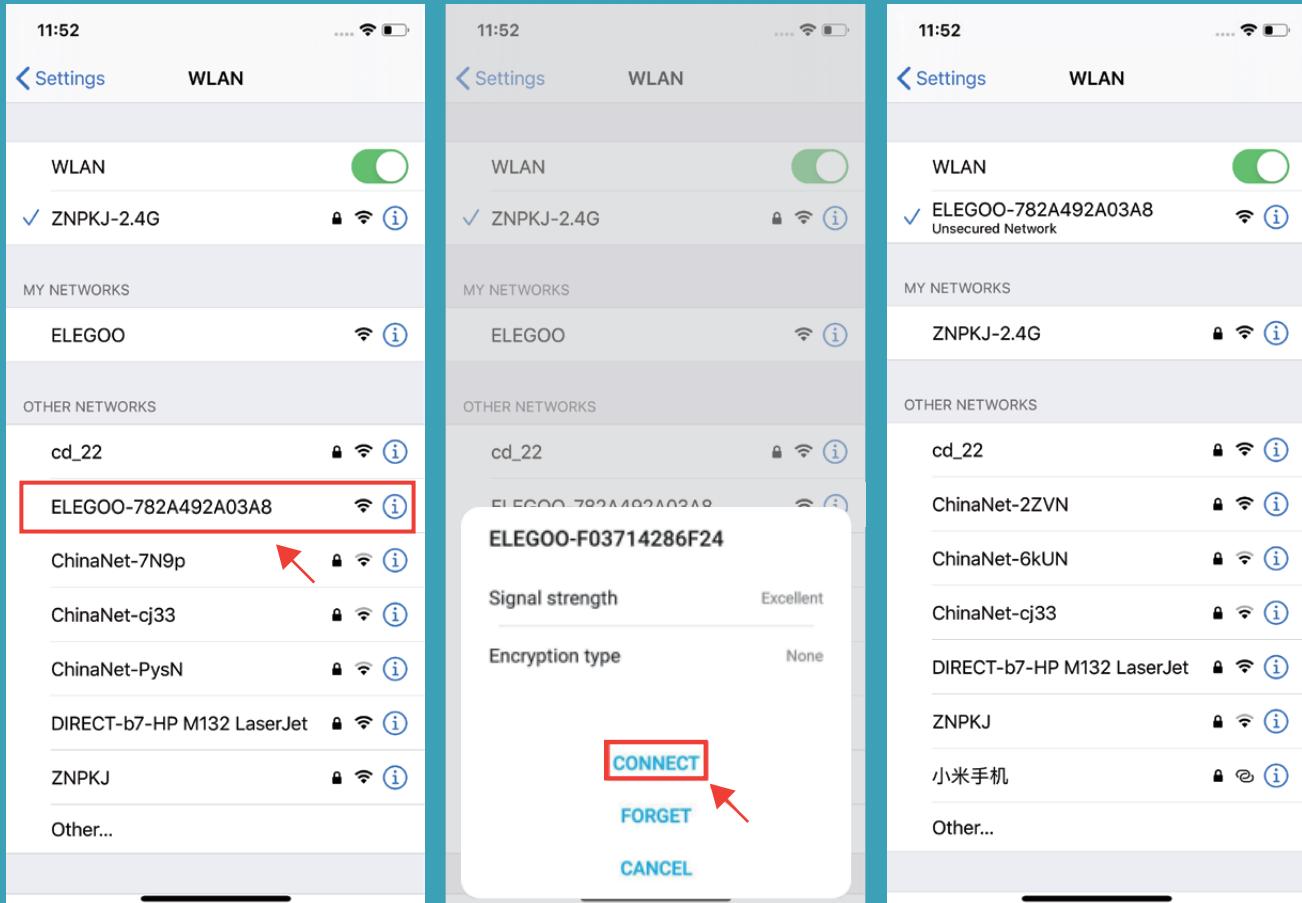


1 The Use of Graphical Programming

- + 1. First of all, please make sure that the Smart Robot Car 4.0 has uploaded the program successfully. If not, please open the **SmartRobotCarV4.0** file in the folder and then click the program **SmartRobotCarV4.0.ino** to upload. After uploading successfully, please turn the “**Upload-Cam**” button to “**Cam**” and then turn on the power of the car.



- + 2.Turn on the phone to connect to the WiFi hotspot opened by the wifi module on the car. (As shown in the figure, the name of the car' s WiFi hotspot would be :ELEGOO-XXXXXXXXXXXXXX(a string of numbers), this string of numbers is a unique ID of the car, just like our ID card number.)



- + 3.After the WiFi has been connected successfully, we can open our supporting APP.



EleRobot

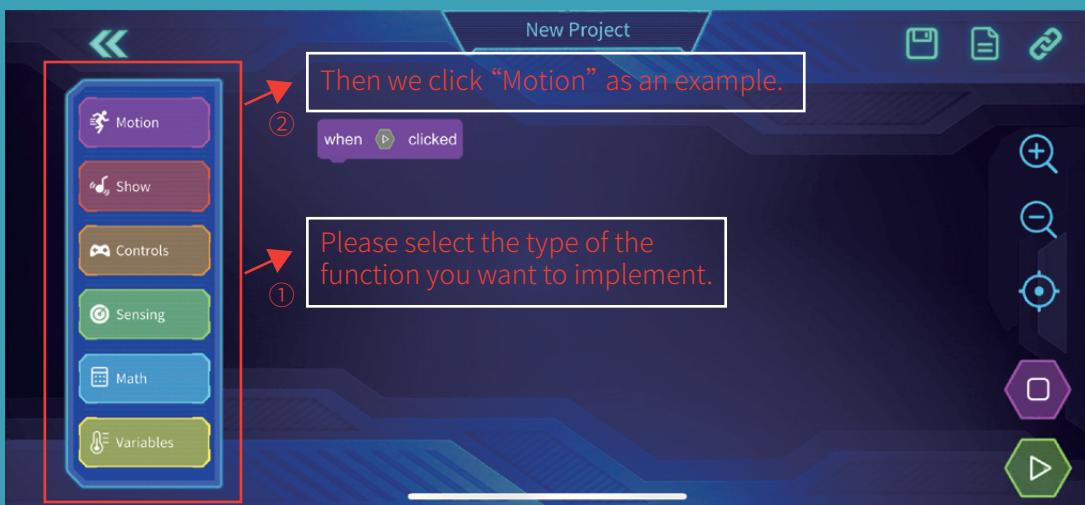
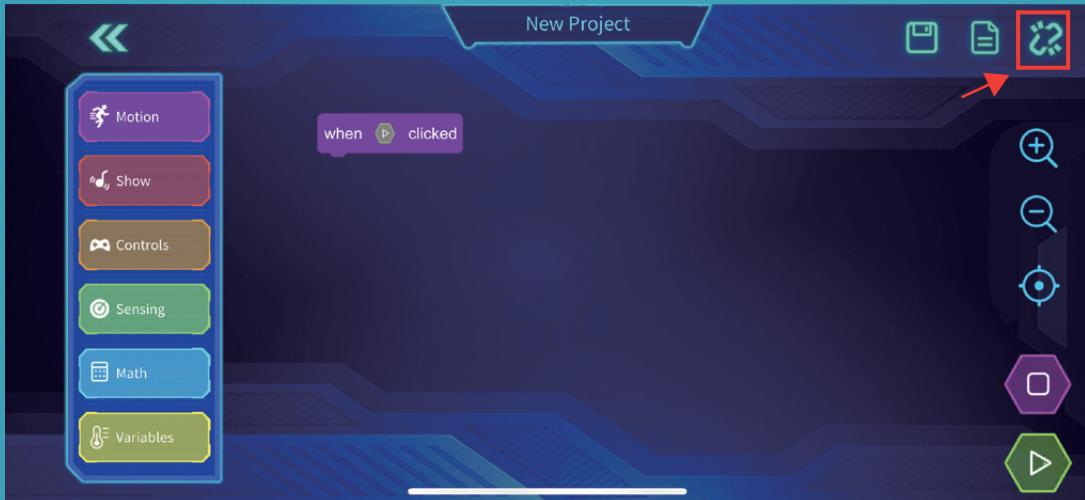
- + 4.Enter the menu interface, and we need to first make sure that the current interface is the control interface of SmartRobotCar .

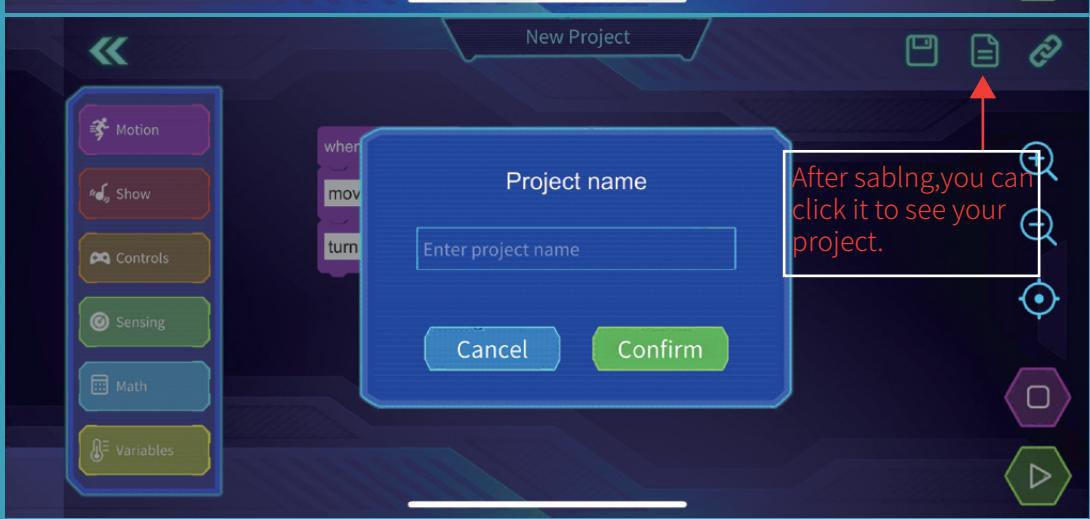
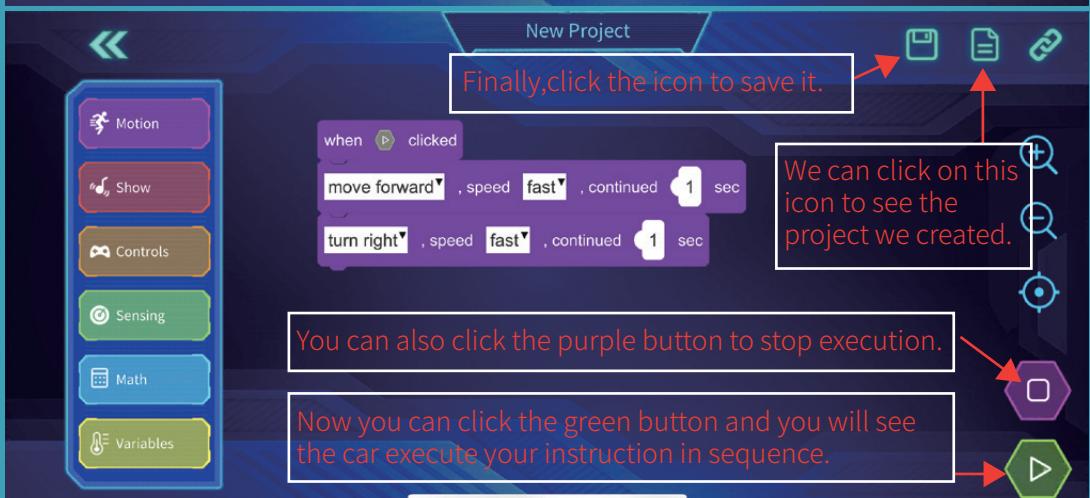
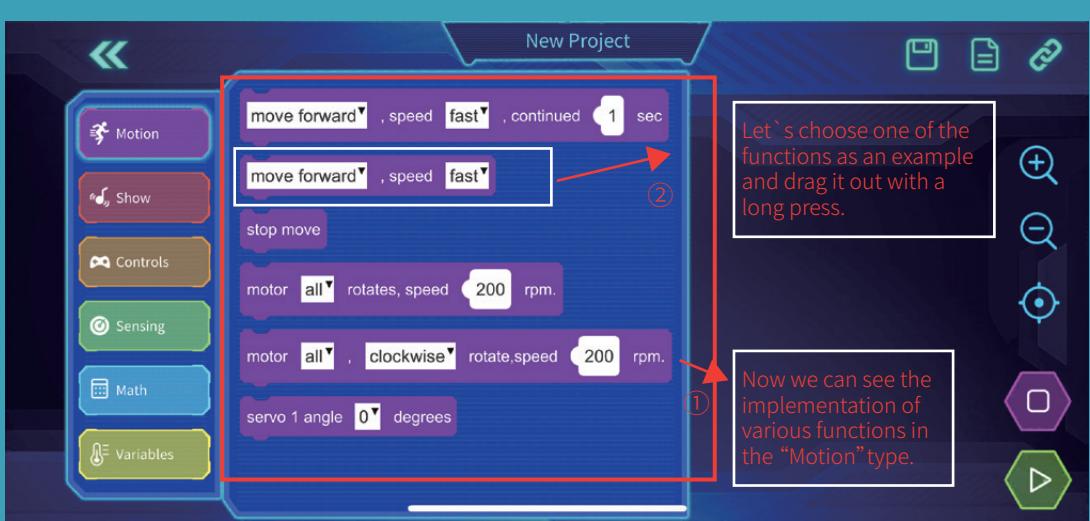


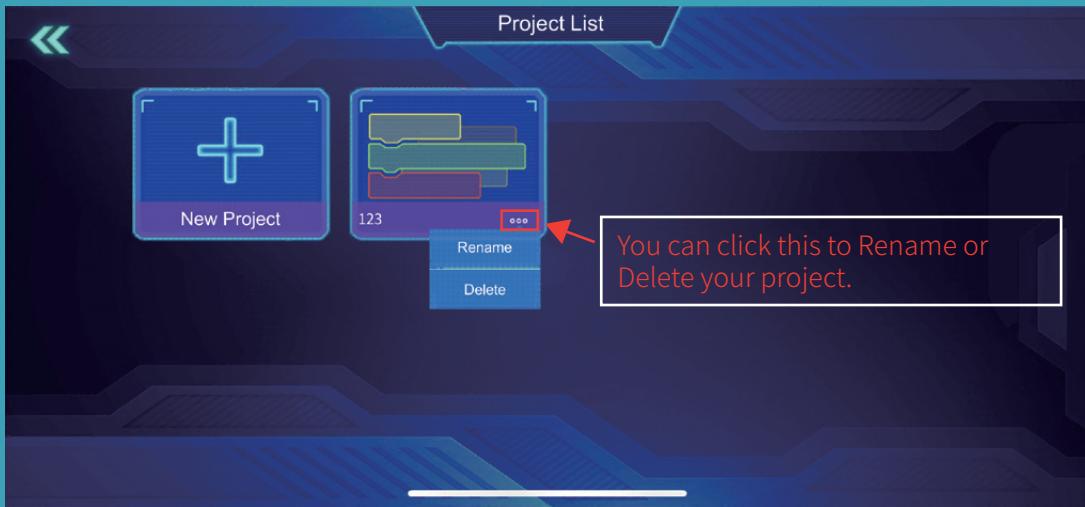
- + 5.Then click "Program" for graphical programming.



+ 6. After entering the "Program" interface, please make sure that the phone is connected to the car successfully. If the connection status icon in the upper right corner is disconnected, please click this icon to connect first. After the connection is successful, you can proceed with specific operations.







2 DIY Mode

+ In the DIY mode, we can design our own buttons to send customized messages to the car. At the same time, we can also write a program through Arduino software and upload it to the car according to our customized information, so that the car can perform corresponding operations based on the messages we send.

+ 1. Select “DIY” on the menu interface. (First of all, you need to make sure that WiFi is connected.)

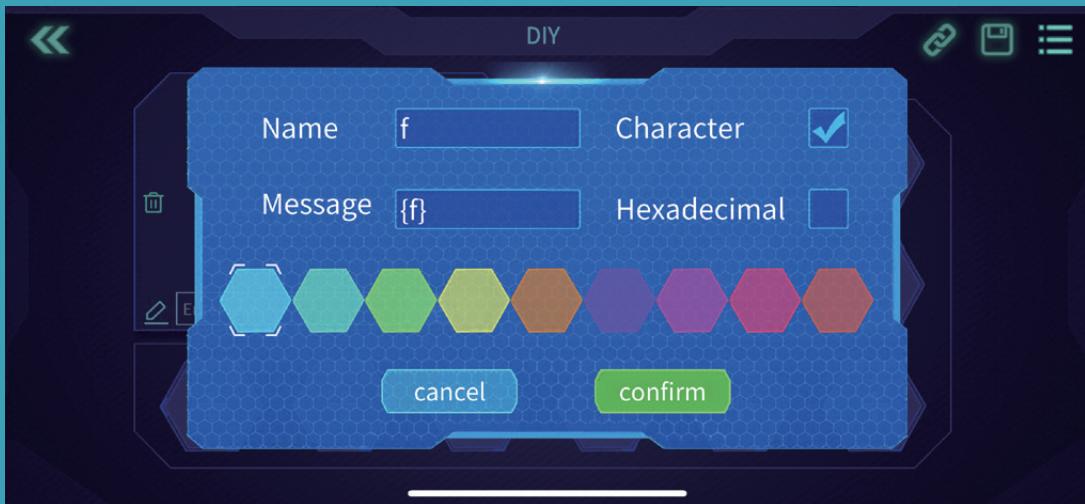




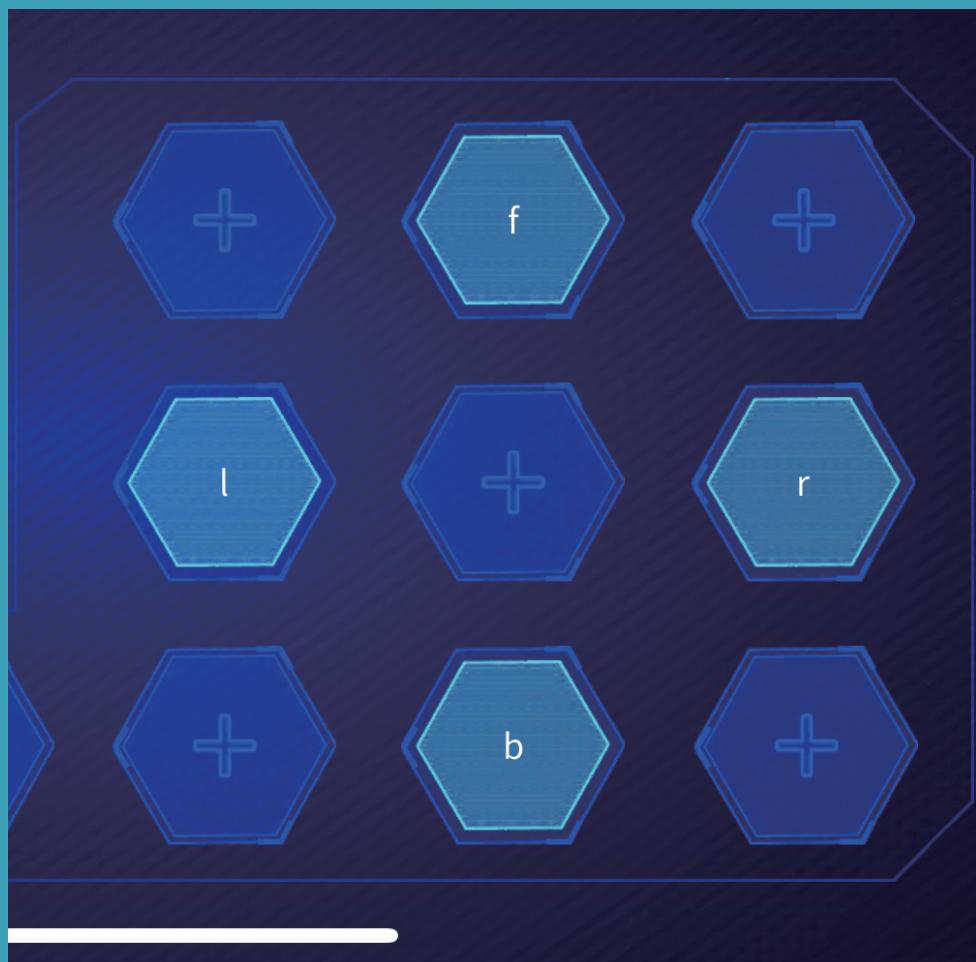
⊕ 2. If you have uploaded the complete **SmartRobotCarV4.0.ino** program, then you could verify the settings as follows.

We set the Message format as "`{" + "Message" + "}"`"

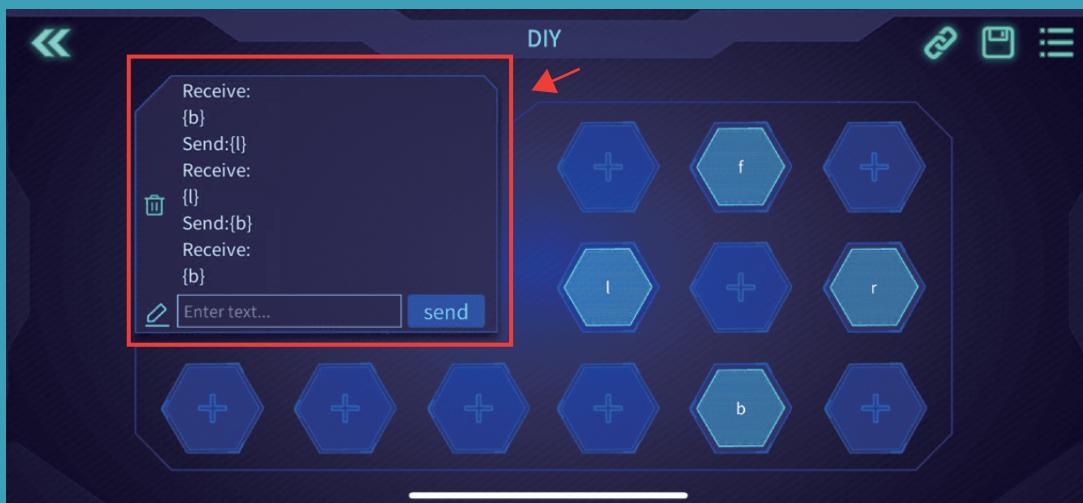
And you can choose to send the messages in characters or hexadecimal numbers.



 Take the above figure as an example, the other keys are set in turn as follows:



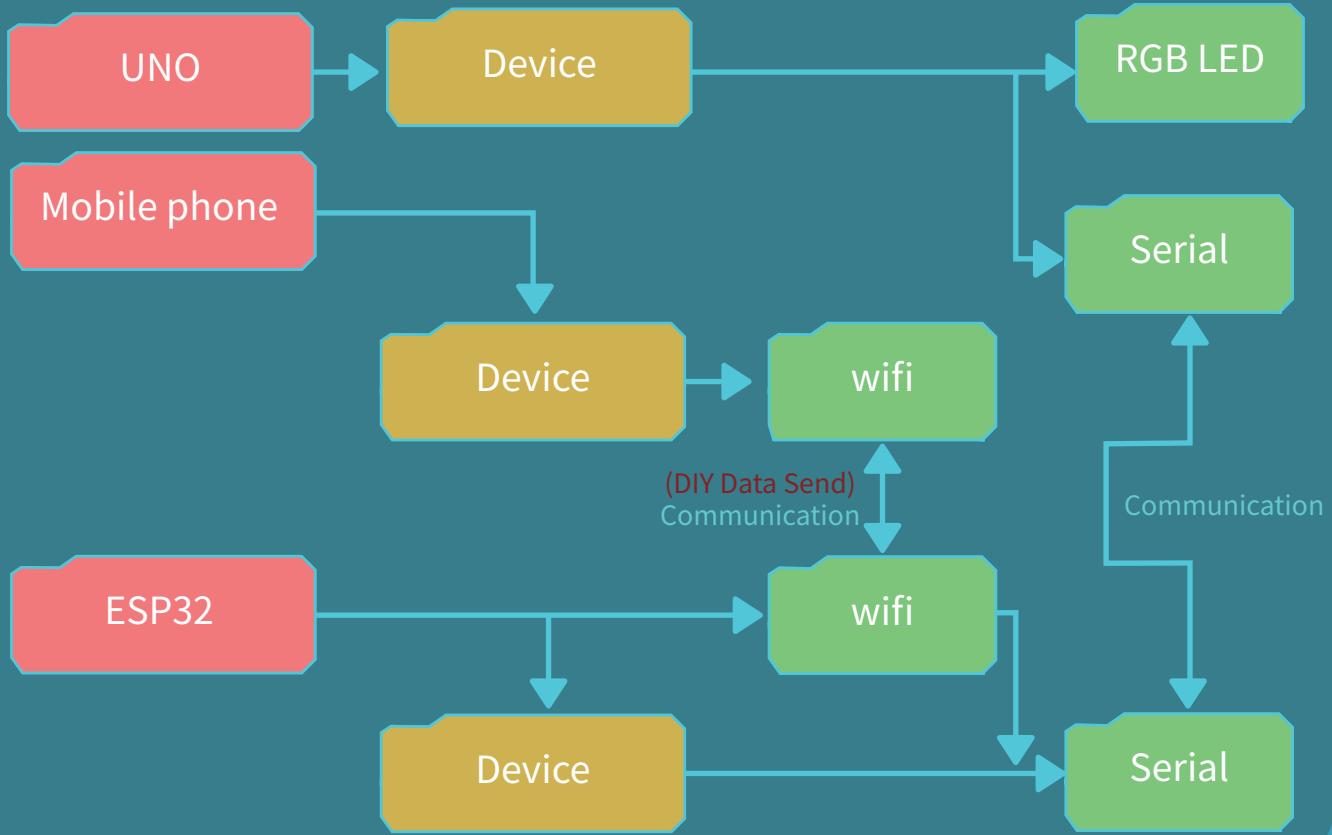
 When the setting is finished, please press the keys in sequence, and if we receive the return message (as shown in the following figure), it means the DIY setting is successful.



+ 3. Now, let's program a simple example depends on the DIY function.

The specific implementation chart is as follows:

Implementation principle of DIY Demo1



+ 4. First of all, let's complete part of the programming of the car.
Please open **Demo 1** in the current folder:

+ 5. We need to process the data {Message} received by the serial port to get Message.

This return value is the parsed command information.



```
C:/ int SerialPortDataAnalysis(void)
```

+ `Serial.available()`:return to the serial buffer, the current number of characters remaining in the buffer of the serial port. This function is generally used to determine if there is data in the buffer of the serial port.

+ `Serial.read()`: take out and read a Byte data from the buffer of the serial port.

+ `int atoi(const char *str)`:Converts the string pointed by the parameter str to an integer (int type).

+ `s.substr(pos, end)`:Intercepts the character string substring from pos (including 0) to end in the string s, and returns.

+ `c_str()`:Converts the string object to a string style in C.

+ 6.After we parse out the key command information, we can operate our car to perform different operations based on different information.

In our example in this lesson, we change the color of the colored lights on the car by sending different information: {1}, {2}, {3} and {4}.

If you don't know how to change the color of the light, you can go back and read the `Demo5` review in Lesson 7 to find out.

Of course, you can also make the car achieve the function you want according to your own ideas.



C:/ void Event()

+ 7.After the program is written, turn the “Upload-Cam” button to “Upload” to upload the program. And turn it back to “Cam” after the program has been uploaded successfully and then take out your phone to connect to WiFi.

+ 8.Finally, open the APP to enter the DIY mode. First, you need to make sure the WiFi  is connected, and then set the button as shown in the figure bellow.

Click the button 1, 2, 3 and 4 to send message {1}, {2}, {3} and {4}. The car will change the color of the LED light according to the messages sent by different buttons.

